Aufgabe 7: josh (14 Punkte) Bearbeitung in Zweier-Gruppen

Programmieren Sie basierend auf der clash eine um Signalbehandlung, Job-Verwaltung und E/A-Umleitung erweiterte Shell: **josh** (**jo**b **sh**ell). Verwenden Sie als Ausgangsbasis dafür unbedingt die Vorgaben in **/proj/i4sp/pub/aufgabe7/**. Die unten gestellten Fragen sind in **doc/josh.txt** zu erläutern.

a) Sofortiges Aufsammeln von Zombieprozessen (RCS Release # 1)

Die clash sammelt angefallene Zombieprozesse jeweils vor der Ausgabe eines Promptsymbols auf. Ändern Sie dieses Verhalten so, dass Zombieprozesse direkt nach deren Entstehen aufgesammelt werden (**sigaction(2)**). Die Ausgabe des Exitstatus und der Kommandozeile kann unmittelbar nach Aufsammeln des Zombieprozesses auf dem Standardfehlerkanal erfolgen.

Legen Sie ein **Unterverzeichnis für RCS** an und checken Sie diese Version von **josh.c** als Release 1 ein (**ci -u josh.c**). Am Ende jeder weiteren Teilaufgabe soll nun das in Klammern angegebene Release eingecheckt werden. Pro Release können beliebig viele Level eingecheckt werden - der jeweils aktuellste Level jedes Release repräsentiert die Abgabe der jeweiligen Teilaufgabe. Sollten Sie einen Fehler bemerken, der sich auch in früheren Releases auswirkt, so muss dieser **nicht** in jedem vorherigen Release ausgebessert werden. Maßgeblich für die Bewertung ist der Endstand Ihrer Shell.

b) Signalhandler (RCS Release # 2)

josh soll nun das INT-Signal vom Terminal (**CTRL+C**) behandeln. In der Signalbehandlungsfunktion soll nur die Meldung "Interrupt!" auf den Standardfehlerkanal ausgegeben werden. Was passiert, wenn Ihr josh-Prozess ein Interrupt-Signal erhält und nur ein Vordergrundprozess läuft bzw. wenn auch Hintergrundprozesse laufen (☞ Doku)? Ändern Sie das Programm nun so, dass die Hintergrundprozesse das INT-Signal **ignorieren**. Was ändert sich dadurch am Verhalten bei einem **CTRL+C** an das Terminal (☞ Doku)?

c) Jobverwaltung (RCS Release # 3)

Implementieren Sie in der Shell ein Kommando **jobs**, das die Kommandozeile und Prozess-Id aller laufenden Hintergrundprozesse ausgibt. Zur Verwaltung der Hintergrundprozesse (Jobs) verwenden Sie bitte die Implementierung aus **/proj/i4sp/pub/aufgabe7/plist.[ch]**.

d) Nebenläufigkeitsprobleme (RCS Release #3 oder bereits in früheren Releases enthalten)

Durch die nebenläufige Arbeit auf der Jobliste durch Signalbehandlungen und den eigentlichen Programmablauf kann es zu sog. *Race Conditions* kommen. Identifizieren Sie die möglichen Probleme und Lösungen und beschreiben Sie diese in der Dokumentation (**sigprocmask(2)**).

e) Umleitung der Standard Ein- und Ausgabe für Kindprozesse (RCS Release # 3 oder # 4)

Zuletzt soll die Shell noch das Umleiten der Standardein- und -ausgabe erlauben (**dup2(2)**). Die Standardeingabe wird aus einer Datei *file* gelesen, wenn das Token *<file* in der Kommandozeile auftritt, die Standardausgabe wird in eine Datei *file* umgeleitet, wenn das Token *>file* auftritt, wobei *file* angelegt wird, falls dieses noch nicht existiert, und überschrieben wird, wenn es bereits existiert. Die Umleitzeichen < bzw. > sind hierbei immer mit dem Dateinamen verbunden. Die Token sollen - soweit verwendet - immer am Ende der Kommandozeile in der Reihenfolge <, > und & auftreten.

Hinweis: Wenn Sie die sigaction-Option **SA_RESTART** verwenden möchten, müssen Sie die zusätzliche Compileroption **-D_XOPEN_SOURCE=500** verwenden. Ihre Shell soll mit Optimierungsstufe **-O3** des gcc-Compilers übersetzt werden und funktionieren.

Abgabe: bis spätestens Dienstag, 19.01.2010, 17:30 Uhr